fc	fractional capability variable
x, g, a, b	expression variable
k	integer variable
el	array-element variable

symb	::=			
	|	λ		
	|	$\otimes$		
	|	$\multimap$		
	|	$\vdash$		
	|	$\in$		
	|	$\forall$		
	|	Cap		
	|	Type		
	|	!		
	|	$\rightarrow$		
	|	$\neq$		
f	::=			fractional capability
	|	fc		variable
	|	**Z**		zero
	|	**S** f		successor
t	::=			linear type
	|	**unit**		unit
	|	**bool**		boolean (true/false)
	|	**int**		63-bit integers
	|	**elt**		array element
	|	f **arr**		arrays
	|	$!t$		multiple-use type
	|	$\forall\, fc.t$	bind fc in t	frac. cap. generalisation
	|	$t \otimes t'$		pair
	|	$t \multimap t'$		linear function
	|	$t\{f/fc\}$	M	substitution
e	::=			expression
	|	p		primitives (arithmetic, L1 BLAS, Owl)
	|	x		variable
	|	()		unit introduction
	|	**true**		true (boolean introduction)
	|	**false**		false (boolean introduction)
	|	**if** e **then** e_1 **else** e_2		if (boolean elimination)
	|	k		integer
	|	el		array element
	|	**many** e		packing-up a non-linear value
	|	**let many** $x = e$ **in** e'		using a non-linear value
	|	**fun** $fc \rightarrow e$		frac. cap. abstraction
	|	$e[f]$		frac. cap. specialisation
	|	(e, e')		pair introduction
	|	**let** $(a, b) = e$ **in** e'	bind $a \cup b$ in e'	pair elimination
	|	**fun** $x : t \rightarrow e$	bind x in e	abstraction
	|	$e\ e'$		application

p	::=		primitive
	|	**fix**	fixpoint
	|	**set**	array index assignment
	|	**get**	array indexing
	|	$(+)$	integer addition
	|	$(-)$	integer subtraction
	|	$(*)$	integer multiplication
	|	$(/)$	integer division
	|	$(=)$	integer equality
	|	$(<)$	integer less-than
	|	$(+.)$	element addition
	|	$(-.)$	element subtraction
	|	$(*.)$	element multiplication
	|	$(/.)$	element division
	|	$(= .)$	element equality
	|	$(< .)$	element comparsion (less-than)
	|	$(\&\&)$	boolean conjuction
	|	$(\|\|)$	boolean disjunction
	|	**not**	boolean negation
	|	**share**	share array
	|	**unshare**	unshare array
	|	**free**	free arrary
	|	**array**	Owl: make array
	|	**copy**	Owl: copy array
	|	**sin**	Owl: sine of all elements in array
	|	**hypot**	Owl: $x_i := \sqrt{x_i^2 + y_i^2}$
	|	**mapi**	Owl: $x_i := f(i, x_i)$
	|	**asum**	BLAS: $\sum_i \lvert x_i \rvert$
	|	**axpy**	BLAS: $x := \alpha x + y$
	|	**dot**	BLAS: $x \cdot y$
	|	**rotmg**	BLAS: gen. mod. Givens rotation
	|	**scal**	BLAS: $x := \alpha x$
	|	**amax**	BLAS: index of maximum absolute value

Θ	::=		fractional capability environment
	|	.	
	|	Θ, fc	

Γ	::=		linear types environment
	|	.	
	|	$\Gamma, x : t$	
	|	Γ, Γ'	

Δ	::=		linear types environment
	|	.	
	|	$\Delta, x : t$	

formula ::=
- | *judgement*
- | $x : t \in \Delta$
- | $x : t \in \Gamma$
- | $fc \in \Theta$
- | $t \neq t'$

Well_Formed ::=
- | $\Theta \vdash f\ \mathsf{Cap}$ Valid fractional capabilities
- | $\Theta \vdash t\ \mathsf{Type}$ Valid types

Types ::=
- | $\Theta; \Delta; \Gamma \vdash e : t$ Tying rules for expressions (no primitives yet)

judgement ::=
- | *Well_Formed*
- | *Types*

user_syntax ::=
- | *fc*
- | *x*
- | *k*
- | *el*
- | *symb*
- | *f*
- | *t*
- | *e*
- | *p*
- | Θ
- | Γ
- | Δ
- | *formula*

$\boxed{\Theta \vdash f\ \mathsf{Cap}}$ Valid fractional capabilities

$$\frac{fc \in \Theta}{\Theta \vdash fc\ \mathsf{Cap}}\ \text{WF\_Cap\_Var}$$

$$\frac{}{\Theta \vdash \mathbf{Z}\ \mathsf{Cap}}\ \text{WF\_Cap\_Zero}$$

$$\frac{\Theta \vdash f\ \mathsf{Cap}}{\Theta \vdash \mathbf{S}\ f\ \mathsf{Cap}}\ \text{WF\_Cap\_Succ}$$

$\boxed{\Theta \vdash t\ \mathsf{Type}}$ Valid types

$$\frac{}{\Theta \vdash \mathbf{unit}\ \mathsf{Type}}\ \text{WF\_Type\_Unit}$$

$$\frac{}{\Theta \vdash \mathbf{bool}\ \mathsf{Type}}\ \text{WF\_Type\_Bool}$$

$$\frac{}{\Theta \vdash \mathbf{int}\ \mathsf{Type}}\ \text{WF\_Type\_Int}$$

$$\frac{}{\Theta \vdash \mathbf{elt}\ \mathsf{Type}}\ \text{WF\_Type\_Elt}$$

$$\frac{\Theta \vdash f\ \mathsf{Cap}}{\Theta \vdash f\ \mathbf{arr}\ \mathsf{Type}}\ \textsc{WF\_Type\_Array}$$

$$\frac{\Theta \vdash t\ \mathsf{Type}}{\Theta \vdash\ !t\ \mathsf{Type}}\ \textsc{WF\_Type\_Bang}$$

$$\frac{\Theta, fc \vdash t\ \mathsf{Type}}{\Theta \vdash \forall fc.t\ \mathsf{Type}}\ \textsc{WF\_Type\_Gen}$$

$$\frac{\Theta \vdash t\ \mathsf{Type} \quad \Theta \vdash t'\ \mathsf{Type}}{\Theta \vdash t \otimes t'\ \mathsf{Type}}\ \textsc{WF\_Type\_Pair}$$

$$\frac{\Theta \vdash t\ \mathsf{Type} \quad \Theta \vdash t'\ \mathsf{Type}}{\Theta \vdash t \multimap t'\ \mathsf{Type}}\ \textsc{WF\_Type\_Lolly}$$

$\boxed{\Theta; \Delta; \Gamma \vdash e : t}$ Tying rules for expressions (no primitives yet)

$$\frac{}{\Theta; \Delta; \cdot, x : t \vdash x : t}\ \textsc{Ty\_Var\_Lin}$$

$$\frac{x : t \in \Delta}{\Theta; \Delta; \cdot \vdash x : t}\ \textsc{Ty\_Var}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash () : !\mathbf{unit}}\ \textsc{Ty\_Unit\_Intro}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \mathbf{true} : !\mathbf{bool}}\ \textsc{Ty\_Bool\_True}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \mathbf{false} : !\mathbf{bool}}\ \textsc{Ty\_Bool\_False}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e : \mathbf{bool} \quad \Theta; \Delta; \Gamma' \vdash e_1 : t' \quad \Theta; \Delta; \Gamma' \vdash e_2 : t'}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{if}\ e\ \mathbf{then}\ e_1\ \mathbf{else}\ e_2 : t}\ \textsc{Ty\_Bool\_Elim}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash k : !\mathbf{int}}\ \textsc{Ty\_Int\_Intro}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash el : !\mathbf{elt}}\ \textsc{Ty\_Elt\_Intro}$$

$$\frac{t \neq f\ \mathbf{arr} \quad \Theta; \Delta; \cdot \vdash e : t}{\Theta; \Delta; \cdot \vdash \mathbf{many}\ e : !t}\ \textsc{Ty\_Bang\_Intro}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e : !t \quad \Theta; \Delta, x : t; \Gamma' \vdash e' : t'}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let\ many}\ x = e\ \mathbf{in}\ e' : t'}\ \textsc{Ty\_Bang\_Elim}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e : t \quad \Theta; \Delta; \Gamma' \vdash e' : t'}{\Theta; \Delta; \Gamma, \Gamma' \vdash (e, e') : t \otimes t'}\ \textsc{Ty\_Pair\_Intro}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e_{12} : t_1 \otimes t_2 \quad \Theta; \Delta; \Gamma', a : t_1, b : t_2 \vdash e : t}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let}\ (a, b) = e_{12}\ \mathbf{in}\ e : t}\ \textsc{Ty\_Pair\_Elim}$$

$$\frac{\Theta \vdash t' \,\mathsf{Type} \quad \Theta; \Delta; \Gamma, x : t' \vdash e : t}{\Theta; \Delta; \Gamma \vdash \mathbf{fun}\, x : t' \to e : t' \multimap t} \;\textsc{Ty\_Lambda}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e : t' \multimap t \quad \Theta; \Delta; \Gamma' \vdash e' : t'}{\Theta; \Delta; \Gamma, \Gamma' \vdash e\, e' : t} \;\textsc{Ty\_App}$$

$$\frac{\Theta, fc; \Delta; \Gamma \vdash e : t}{\Theta; \Delta; \Gamma \vdash \mathbf{fun}\, fc \to e : \forall\, fc.t} \;\textsc{Ty\_Gen}$$

$$\frac{\Theta \vdash f \,\mathsf{Cap} \quad \Theta; \Delta; \Gamma \vdash e : \forall\, fc.t}{\Theta; \Delta; \Gamma \vdash e[f] : t\{f/fc\}} \;\textsc{Ty\_Spc}$$

```
Definition rules:         19 good    0 bad
Definition rule clauses: 43 good    0 bad
```